**PCT** WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04L 9/32, G06F 12/14

A1

(11) International Publication Number: **WO 98/15086**

(43) International Publication Date: 9 April 1998 (09.04.98)

(21) International Application Number: PCT/US97/13518

(22) International Filing Date: 30 July 1997 (30.07.97)

(30) Priority Data:
08/722,298 30 September 1996 (30.09.96) US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventor: DAVIS, Derek, L.; 4509 E. Desert Trumpet Road, Phoenix, AZ 85044 (US).

(74) Agents: TAYLOR, Edwin, H. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

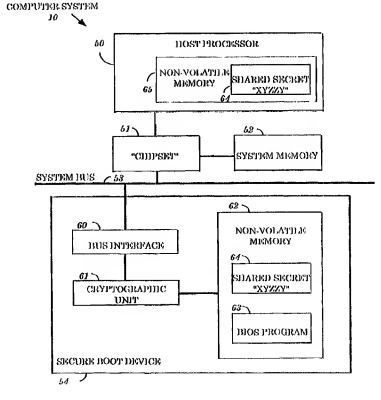
(81) Designated States: AL, AM, AT, AT (Utility model), AU (Petty patent), AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

**Published**
*With international search report.*

(54) Title: SECURE BOOT

(57) Abstract

A subsystem prevents unauthorized replacement of boot-up firmware (e.g., BIOS (63)) embedded in modifiable non-volatile memory devices (620) such as flash memory. The firmware device is contained in a secure boot device (54) which is responsive to the host processor (50). The security protection is established by the encryption and decryption of the boot-up instructions using a secret key (64) shared by both the secure boot device (54) and the host processor (50).



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **AL** | Albania | **ES** | Spain | **LS** | Lesotho | **SI** | Slovenia |
| **AM** | Armenia | **FI** | Finland | **LT** | Lithuania | **SK** | Slovakia |
| **AT** | Austria | **FR** | France | **LU** | Luxembourg | **SN** | Senegal |
| **AU** | Australia | **GA** | Gabon | **LV** | Latvia | **SZ** | Swaziland |
| **AZ** | Azerbaijan | **GB** | United Kingdom | **MC** | Monaco | **TD** | Chad |
| **BA** | Bosnia and Herzegovina | **GE** | Georgia | **MD** | Republic of Moldova | **TG** | Togo |
| **BB** | Barbados | **GH** | Ghana | **MG** | Madagascar | **TJ** | Tajikistan |
| **BE** | Belgium | **GN** | Guinea | **MK** | The former Yugoslav Republic of Macedonia | **TM** | Turkmenistan |
| **BF** | Burkina Faso | **GR** | Greece | | | **TR** | Turkey |
| **BG** | Bulgaria | **HU** | Hungary | **ML** | Mali | **TT** | Trinidad and Tobago |
| **BJ** | Benin | **IE** | Ireland | **MN** | Mongolia | **UA** | Ukraine |
| **BR** | Brazil | **IL** | Israel | **MR** | Mauritania | **UG** | Uganda |
| **BY** | Belarus | **IS** | Iceland | **MW** | Malawi | **US** | United States of America |
| **CA** | Canada | **IT** | Italy | **MX** | Mexico | **UZ** | Uzbekistan |
| **CF** | Central African Republic | **JP** | Japan | **NE** | Niger | **VN** | Viet Nam |
| **CG** | Congo | **KE** | Kenya | **NL** | Netherlands | **YU** | Yugoslavia |
| **CH** | Switzerland | **KG** | Kyrgyzstan | **NO** | Norway | **ZW** | Zimbabwe |
| **CI** | Côte d'Ivoire | **KP** | Democratic People's Republic of Korea | **NZ** | New Zealand | | |
| **CM** | Cameroon | | | **PL** | Poland | | |
| **CN** | China | **KR** | Republic of Korea | **PT** | Portugal | | |
| **CU** | Cuba | **KZ** | Kazakstan | **RO** | Romania | | |
| **CZ** | Czech Republic | **LC** | Saint Lucia | **RU** | Russian Federation | | |
| **DE** | Germany | **LI** | Liechtenstein | **SD** | Sudan | | |
| **DK** | Denmark | **LK** | Sri Lanka | **SE** | Sweden | | |
| **EE** | Estonia | **LR** | Liberia | **SG** | Singapore | | |

# SECURE BOOT

## CROSS-REFERENCES TO RELATED APPLICATION

The named inventor of the present application has previously filed a United States Patent Application entitled "An Apparatus and Method for Cryptographic Companion Imprinting", filed December 4, 1995 (Application No. 08/566,910). This Application is owned by the same Assignee of the present Application.

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

This invention relates to the field of security of computer firmware, especially in the areas of boot-up firmware including Operating System ("OS") and Basic Input and Output System ("BIOS") in general computing systems, in particular personal computers ("PCs").

### 2. Description of Related Art

One of the most critical elements in a computer system is boot-up firmware. The boot-up firmware may be an Operating System ("OS"), a portion of the OS, or the Basic Input and Output System ("BIOS"). The boot-up firmware is essentially the machine code typically stored in some form of non-volatile memory to allow a Central Processing Unit ("CPU") to perform tasks such as initialization, diagnostics, loading the operating system kernel from mass storage, and routine input/output ("I/O") functions.

Upon initially supplying power to the CPU through a power-up sequence, the CPU will "boot up" by fetching the instruction code residing in the boot-up firmware. Traditionally, the boot-up firmware is implemented in Erasable Programmable Read

Only Memory ("EPROM"). However, recent advances in semiconductor technology have allowed boot-up firmware to be implemented in flash memory, increasing its susceptibility to intrusive attack.

Due to its critical role in computer systems, boot-up firmware should be well protected against intrusive attacks. One type of intrusive attack involves an intruder accessing the computer directly, physically removing a boot-up device containing the boot-up firmware (e.g., flash memory, a printed circuit board containing memory, etc.), and substituting that boot-up device with another boot-up device. In some cases, the intruder may be the legitimate owner or user of the computer system who is trying to defraud third-party service providers.

Currently, mechanical security mechanisms, particularly those used by portable computers to erase important information if the laptop's casing is opened without authorization, has little effect in preventing these intrusive attacks. There are no well-established electronic security mechanisms to provide security protection for a path connecting a host processor and a boot-up device.

Therefore, it would be desirable to design a mechanism that prevents an intruder from successfully defrauding others through replacement of the boot-up device such as a cryptographic coprocessor, or a flash memory device for example. This may be achieved by electrically "binding" the physical boot-up device to the host processor to provide a secure path between the host processor and the boot-up firmware. This prevents an attacker from simply replacing the cryptographic coprocessor, since the host processor is unable to execute boot-up instructions not previously encrypted by the specific cryptographic coprocessor to which it has been "imprinted".

## SUMMARY OF THE INVENTION

The present invention describes a secure subsystem to prevent unauthorized replacement of a storage device containing a boot-up executable code by establishing a

secure path between a secure boot device and a host processor based on an electronic keying mechanism.

The secure boot device is coupled to the storage device and encrypts the executable code based on a secret key to generate an encrypted code. The host processor then decrypts the encrypted code based on the same secret key to generate a decrypted code. The host processor executes the decrypted code only if the decrypted code corresponds to the executable code. A communication path is established between the secure boot device and the host processor to allow the two processors to communicate securely by exchanging such encrypted messages.

## BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

**FIG. 1** is a diagram showing the present invention with a secure path between the host processor and the secure boot device, thus enabling a secure boot of the system.

**FIG. 2** is a flowchart of the operations that occur in the present invention during a normal read access to the boot-up program by the host processor.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention provides a secure path between a host processor and a memory device containing a boot-up program by establishing a secure communication protocol between the host processor and a secure boot device. In the following description, some terminology is used to discuss certain cryptographic features. For example, a "key" is an encoding and/or decoding parameter used by conventional cryptographic algorithms such as Rivest, Shamir and Adleman ("RSA"), Data Encryption Algorithm ("DEA") as specified in Data Encryption Standard ("DES") and the like. A "secret key" is a key used for both encryption and decryption by a limited number of electronic devices having access to that key.

As described below, the secure boot device responds to the requests from a host processor ("host requests") for accessing a boot-up program by encrypting the instruction code in the boot-up program using a secret key shared with the host processor. The encrypted instruction code is decrypted by the host processor using the same secret key. Since the secret key is known only by the host processor and the secure boot device, any attempt to replace the secure boot device containing the boot-up program, will result in incorrect decrypted code making the system inoperable.

Referring to **Figure 1**, an embodiment of a computer system utilizing the present invention is shown. The computer system 10 includes a chipset 51 which

operates as an interface to support communications between host processor 50, system memory 52, and devices coupled to a system bus 53. More specifically, host processor 50 includes logic circuitry (not shown) as well as a small amount of internal non-volatile memory 65 used to contain key information. System memory 52 may include, but is not limited to conventional memory such as various types of random access memory ("RAM"), e.g., DRAM, VRAM, SRAM, etc., as well as memory-mapped I/O devices. System bus 53 may be implemented in compliance with any type of bus architecture including Peripheral Component Interconnect ("PCI") and Universal Serial Bus ("USB") and the like.

One of the devices that may be coupled to the system bus 53 includes a secure boot device 54. Secure boot device 54 comprises a bus interface 60, a cryptographic unit 61 and a local non-volatile memory 62. The bus interface 60 is used to establish an electrical connection to system bus 53. The boot-up program 63 is stored within non-volatile memory 62.

Referring still to **Figure 1**, both host processor 50 and secure boot device 54 are configured to contain a shared secret key 64 in their respective non-volatile memories 65 and 62. Established at manufacture during initialization by the Original Equipment Manufacturer or other system suppliers who produce the host processor and the secure boot device, this shared secret key 64 is used for both encryption and decryption by the secure boot device 54 and the host processor 50. The encryption and decryption can be performed through a variety of techniques including specialized hardware circuits, combination of hardware and software, or specialized accelerators. The sequences followed by the host processor 50 and secure boot device 54 for boot-up access during the system power-up ("boot") sequence are described in **Figure 2**.

Referring now to **Figure 2**, the steps associated with the "boot up" phase of the system are shown. First, in Step 110, the host processor issues a read request for an address corresponding to the boot-up program. The secure boot device detects this boot-up address by having its address space mapped to the corresponding boot-up program (Step 112). Upon detection of the read request, the secure boot device proceeds with encrypting the corresponding boot-up instruction using the shared

secret key (Step 114). In Step 116, the secure boot device responds to the host requests with the encrypted boot-up instruction. In Step 118, upon receiving the encrypted boot-up instruction, the host processor decrypts the encrypted boot-up instruction using the shared secret key. In Step 120, the resulting decrypted boot-up instruction may or may not correspond to a correct instruction depending on whether the system has been tampered with or not. If the system has been tampered with, in Step 130, the decrypted boot-up instruction results in an improper or invalid instruction. It is most likely that the system hangs up because of a number of reasons such as bus error, unrecognized opcode, infinite loop, etc. As a result, the boot-up sequence results in system failure. In Step 140, the decrypted boot-up instruction results in a valid or correct instruction in the boot-up program. The host processor executes the instruction and proceeds with the next boot-up instruction until the entire booting sequence is completed.

The shared secret key is known only to the secure boot device and the host processor, and therefore an attempt to subvert the system by replacing the secure boot device by another device is futile. The reason is that the replacement device cannot communicate with the host processor. An intruder, without knowing the shared secret key, cannot duplicate the cryptographic subsystem. The boot-up firmware is therefore protected from the physical replacement of the boot-up device.

Although the above discussion is directed to the secure path between the host processor and the dedicated secure boot device, it is readily realized that the secure path can be established between any number of subsystems, processors, or devices and any combination thereof. A typical secure path involves a secret key shared by all the devices/ processors, and encryption/decryption algorithms implemented by either hardware, firmware, or software or any combination thereof.

In another embodiment of the invention (not shown), a chipset with secure boot device functionality containing some boot-up code is interfaced with the host processor. This boot-up code may be a sequence of executable instructions. A secret key shared by the chipset and the host processor is used to encrypt and decrypt the boot-up code. A secure path is established as discussed above.

Yet another embodiment (not shown) involves a printed circuit board ("PCB") or a "smart" card such as the PCMCIA containing the boot-up program or some executable or information code. The PCB or smart card may be plugged into any expansion slot on the system mother board, or on any backplane interface bus. A secure boot device is coupled to such a PCB or smart card, responding to the host requests by encrypting the boot-up code using a secret key shared by both the board/card and the host processor. The host processor decrypts the encrypted code using the same secret key. The secure boot device may reside on the same PCB or smart card, or anywhere in the system, such as another separate PCB or smart card. As long as the secure boot device is able to communicate with the host processor by exchanging the encrypted or decrypted boot-up code, any attempt to remove the PCB or smart card and replace with another PCB or smart card without the secret key will result in system inoperation.

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the preferred embodiment, as well as other embodiments of the invention which are apparent to persons skilled in the art to which the invention pertains, are deemed to lie within the spirit and scope of the invention.

## CLAIMS

What is claimed is:

1. A system for preventing unauthorized replacement of a storage element containing an executable code, comprising:

first cryptographic means for encrypting said executable code based on a secret key to generate an encrypted code, said first cryptographic means being coupled to said storage element;

second cryptographic means for decrypting said encrypted code based on said secret key to generate a decrypted code, said second cryptographic means being coupled to said first cryptographic means and being capable of executing said decrypted code if said decrypted code corresponds to said executable code; and

communication means for enabling said first cryptographic means to communicate with said second cryptographic means by exchanging said encrypted code and decrypted code.

2. The system according to claim 1, wherein said first cryptographic means includes a secure boot device.

3. The system according to claim 1, wherein said second cryptographic means includes a host processor.

4. The system according to claim 1, wherein said communication means includes an interface coupled to a bus to allow said first cryptographic means to respond to a request for access from said second cryptographic means.

5. The system according to claim 1, wherein said secret key is accessible to said first cryptographic means and said second cryptographic means.

6. The system according to claim 1, wherein said executable code is an Operating System.

7. The system according to claim 1, wherein said executable code is a Basic Input and Output System.

8. The system according to claim 1 wherein the storage element is a modifiable non-volatile memory element.

9. The system according to claim 8 wherein the modifiable non-volatile memory element is a flash memory.

10. A system for preventing unauthorized replacement of an executable code, comprising:

a first processor for encrypting said executable code based on a secret key to generate an encrypted code, said first processor being coupled to said executable code;

a second processor for decrypting said encrypted code based on said secret key to generate a decrypted code, said second processor being coupled to said first processor and being capable of executing said decrypted code if said decrypted code corresponds to said executable code; and

a communication path for enabling said first processor to communicate with said second processor by exchanging said encrypted code and decrypted code.

11. The system according to claim 10, wherein said first processor is a secure boot device.

12. The system according to claim 10, wherein said second processor is a host processor.

13. The system according to claim 10, wherein said communication path includes an interface coupled to a bus to allow said first processor to respond to a request for access from said second processor.

14. The system according to claim 10, wherein said secret key is accessible to said first processor and said second processor.

15. The system according to claim 10, wherein said executable code is an Operating System.

16. The system according to claim 10, wherein said executable code is a Basic Input and Output System.

17. The system according to claim 10 wherein the storage element is a modifiable non-volatile memory element.

18. The system according to claim 17 wherein the modifiable non-volatile memory element is a flash memory.

19. A method for prevent unauthorized replacement of an executable code contained in a storage element and accessible to a host processor, the method comprising the steps of:

providing a security processor which is coupled to said storage element, said security processor being responsive to said host processor;

encrypting the executable code based on a secret key to produce an encrypted code;

decrypting said encrypted code based on said secret key to produce a decrypted code;

executing said decrypted code if said decrypted code corresponds to said executable code; and

establishing a communication path between said host processor and said security processor to allow said host processor to communicate with said security processor.

20. The method according to claim 19, wherein said communication path includes an interface coupled to a bus to allow said security processor to respond to a request for access from said host processor.

21. The method according to claim 19, wherein said secret key is accessible to said host processor and said security processor.

22. The method of claim 19, wherein said executable code is an Operating System.

23. The method of claim 19, wherein said executable code is a Basic Input and Output System.

24. The method of claim 19, wherein said storage element is a modifiable non-volatile memory element.

25. The method of claim 19, whcrein said security processor is a secure boot device.

26. The method of claim 19 wherein said step of encrypting is performed by said security processor and said step of decrypting is performed by said host processor.

27. The method of claim 24, wherein said modifiable non-volatile memory element is a flash memory.

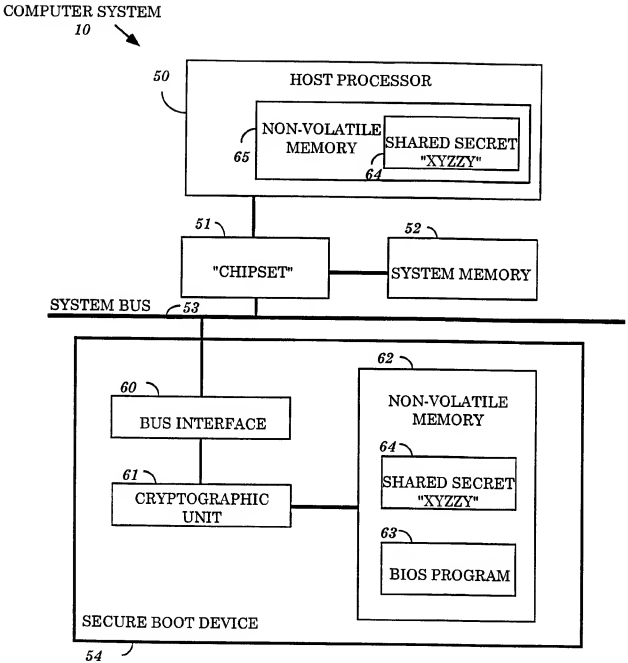

COMPUTER SYSTEM
10
50
HOST PROCESSOR
NON-VOLATILE MEMORY
65
SHARED SECRET "XYZZY"
64
51
"CHIPSET"
52
SYSTEM MEMORY
SYSTEM BUS
53
62
60
BUS INTERFACE
NON-VOLATILE MEMORY
64
SHARED SECRET "XYZZY"
61
CRYPTOGRAPHIC UNIT
63
BIOS PROGRAM
SECURE BOOT DEVICE
54

*FIGURE 1*

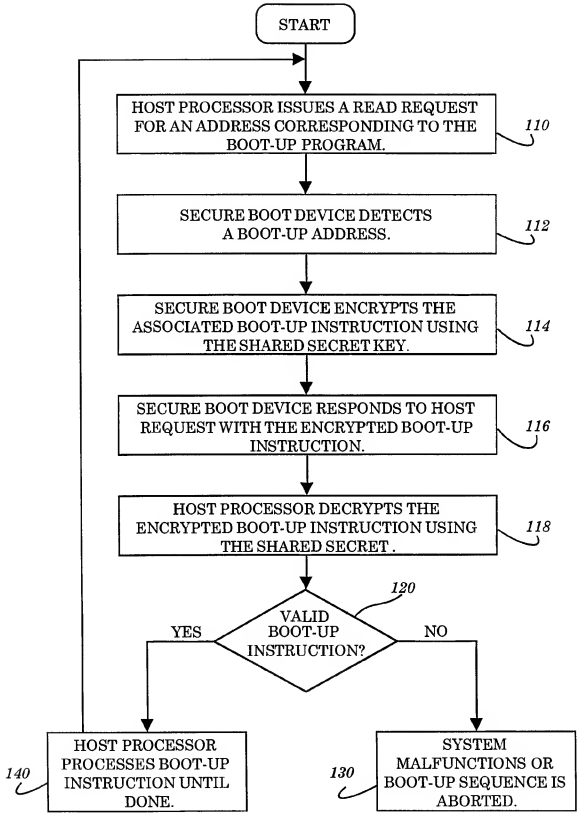

START
HOST PROCESSOR ISSUES A READ REQUEST FOR AN ADDRESS CORRESPONDING TO THE BOOT-UP PROGRAM.
110
SECURE BOOT DEVICE DETECTS A BOOT-UP ADDRESS.
112
SECURE BOOT DEVICE ENCRYPTS THE ASSOCIATED BOOT-UP INSTRUCTION USING THE SHARED SECRET KEY.
114
SECURE BOOT DEVICE RESPONDS TO HOST REQUEST WITH THE ENCRYPTED BOOT-UP INSTRUCTION.
116
HOST PROCESSOR DECRYPTS THE ENCRYPTED BOOT-UP INSTRUCTION USING THE SHARED SECRET .
118
120
VALID BOOT-UP INSTRUCTION?
YES
NO
HOST PROCESSOR PROCESSES BOOT-UP INSTRUCTION UNTIL DONE.
140
SYSTEM MALFUNCTIONS OR BOOT-UP SEQUENCE IS ABORTED.
130

*FIGURE 2*

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/13518

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) : H04L 9/32; G06F 12/14
US CL :380/25, 4
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
U.S. : 380/25, 4, 3; 364/260.81, 949.71; 395/186

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
APS: BOOT UP, ENCRYPT?, OPERATING SYSTEM

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 4,698,617 A (BAUER) 06 October 1987, col. 1, lines 5-20; col. 2, line 60 through col. 3, line 31. | 1-26 |
| A, E | US 5,671,275 A (EZURIKO) 23 September 1997, col. 1, lines 26-38. | 1-26 |
| A | US 5,509,120 A (MERKIN et al) 16 April 1996, col. 2, lines 5-10. | 1-26 |
| A | US 5,386,469 A (YEARSLEY et al) 31 January 1995, col. 2, lines 25-39. | 1-26 |
| A | US 4,764,959 A (WATANABE et al) 16 August 1988, col. 3, lines 43-60. | 1-26 |
| A | US 4,633,388 A (CHIU) 30 December 1986, col. 2, lines 4-33. | 1-26 |

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

| | | | |
|---|---|---|---|
| * | Special categories of cited documents: | "T" | later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention |
| "A" | document defining the general state of the art which is not considered to be of particular relevance | "X" | document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone |
| "E" | earlier document published on or after the international filing date | "Y" | document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| "L" | document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | "&" | document member of the same patent family |
| "O" | document referring to an oral disclosure, use, exhibition or other means | | |
| "P" | document published prior to the international filing date but later than the priority date claimed | | |

| | |
|---|---|
| Date of the actual completion of the international search<br>17 NOVEMBER 1997 | Date of mailing of the international search report<br>24 JAN 1998 |
| Name and mailing address of the ISA/US<br>Commissioner of Patents and Trademarks<br>Box PCT<br>Washington, D.C. 20231<br>Facsimile No. (703) 305-3230 | Authorized officer<br>GILBERTO BARRÓN JR.<br>Telephone No. (703) 306-4177 |

Form PCT/ISA/210 (second sheet)(July 1992)*